

WHAT IS BLOCKCHAIN?

A blockchain is a decentralized, distributed and public digital ledger that is used to record transactions across many computers so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the collusion of the network.

The blockchain concept represents a paradigm shift in how software engineers will write software applications in the future, and it is one of the key concepts that needs to be well understood. We need to really understand five key concepts, and how they interrelate to one another in the context of this new computing paradigm that is unravelling in front of us: the blockchain, decentralized consensus, trusted computing, smart contracts, and proof of work/stake.

How blockchain came into picture?

Blockchain was invented by Satoshi Nakamoto in 2008 to serve as the public transaction ledger of the cryptocurrency **bitcoin**. The **invention** of the **blockchain** for bitcoin made it the first digital currency to solve the double-spending problem without the need of a trusted authority or central server.

The **second innovation** was called blockchain, which was essentially the realization that the underlying technology that operated bitcoin could be separated from the currency and used for all kinds of other interorganizational cooperation.

The **third innovation** was called the “**smart contract**,” embodied in a second-generation blockchain system called ethereum, which built little computer programs directly into blockchain that allowed financial instruments, like loans or bonds, to be represented, rather than only the cash-like tokens of bitcoin.

Smart contracts help you exchange money, property, shares, or anything of value in a transparent, conflict-free way while avoiding the services of a middleman.

The ethereum smart contract platform now has a market cap of around a billion dollars, with hundreds of projects headed toward the market.

Ethereum is an open-source, public, blockchain-based distributed computing **platform** and operating system featuring smart contract (scripting) functionality. ... **Ethereum** was proposed in late 2013 by **Vitalik Buterin**, a cryptocurrency researcher and programmer.

PROBLEM DEFINITION AND SCOPE

Whenever a Customer orders a particular item of his/her interest from a 3rd party, there is an issue of transparency in the mind of customer.

The Customer is not aware about all the entities involved in the whole scenario and also to know about his/her product customer needs to request to the 3rd party vendor

For the respective purpose which in turn gives rise to an issue of a Centralized entity being present in the system.

And also, Customer doesn't have the idea where properly He/she is paying the money and for what thing he is paying the required amount.

There are many shopping web-apps working as a third party in logistics. So, for every order they need to notify particular element of the system. It is somewhat tedious process.

We can use the blockchain to make the system decentralised and also record transactions between parties in a secure and permanent way.

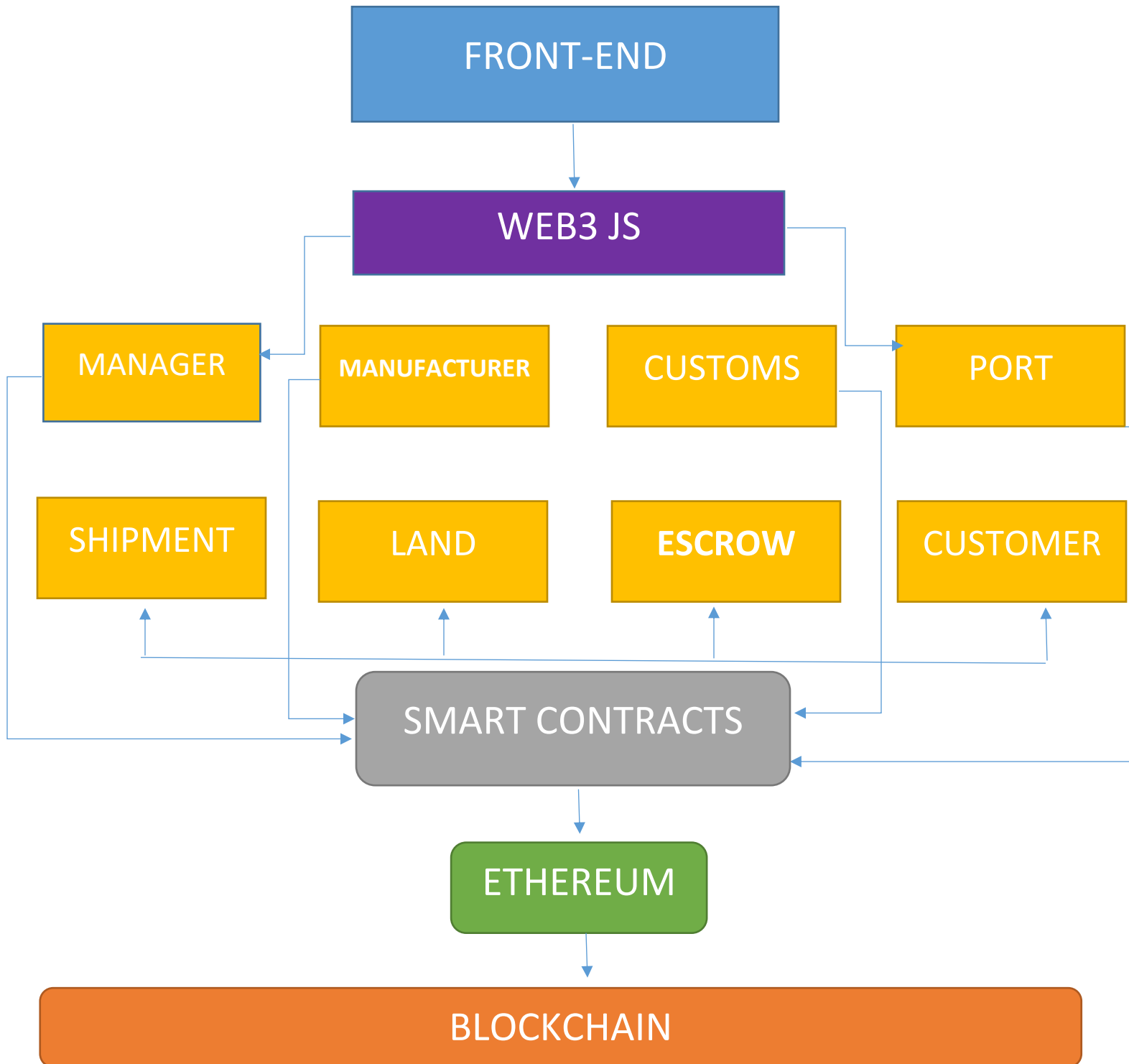
As blockchain is autonomous decentralized system, the issue of transparency is solved as the customer knows about all the entities involved and what is happening in the system with his/her order.

And as the system gives the information about the processes taking place in the system the customer gets the satisfaction for what he is paying for and hence giving the customer and distributor to have a good trust buildup relation.

So, we are bringing the concept of blockchain and smart contracts here where we will define the entities and all the events to be created and generated during the whole order tracking process.

Objective:

- ▶ To develop DAP for tracking the products.
- ▶ To develop SC for International Trade using Blockchain Technology.
- ▶ Increasing the trust between various participants and bringing transparency in the supply chain.
- ▶ Flow of money should be secure and it should use any type of Digital Currency.
- ▶ Using blockchain technology in tracking application can achieve cost savings by more automated, error-free and less paper work processes.
- ▶ To manage the ownership of digital assets and facilitate asset transfers.



IMPLEMENTATION AND RESULT

```

1 pragma solidity ^0.4.19;
2 import "./Ownable.sol";
3 import "./manager.sol";
4 contract Manufacture is Ownable, main
5 {
6     event OrderConfirmed(uint _orderId);
7     event OrderManufactured(uint _orderId);
8     event OrderDispatched(uint _orderId);
9     event DelayInManufacturing(uint _orderId, uint _delayTime);
10
11     function TotaltimeRequired(uint orderId, uint _TotaltimeRequired, uint _expectedTimeOfDeparture) onlyOwner{
12
13         itemMap[orderId].totalTimeRequired = now + _TotaltimeRequired; // Took time required for whole process.
14         statsMap[orderId].timeToNextEntity = now + _expectedTimeOfDeparture; // Took time required to transfer to next entity.
15     }
16
17     function orderConfirmed(uint orderId) onlyOwner payable {
18
19         statsMap[orderId].checkPoint="OrderConfirmed"; // Updates currentStatusOfOrder.
20         statsMap[orderId].timeTheEventCalled=now;
21         OrderConfirmed(orderId); // Event of OrderConfirmed.
22     }
23
24     function orderManufactured(uint orderId) onlyOwner{
25
26         statsMap[orderId].checkPoint="OrderManufactured"; // Updates currentStatusOfOrder.
27         statsMap[orderId].timeTheEventCalled=now;
28         emit OrderManufactured(orderId); // Event of OrderManufactured.
29     }
30
31     function orderDispatched(uint orderId, uint weight) onlyOwner{
32
33         statsMap[orderId].checkPoint="OrderDispatched";
34         statsMap[orderId].timeTheEventCalled=now; // Updates currentStatusOfOrder.
35         itemMap[orderId].Weight=weight;
36         OrderDispatched(orderId); // Event of OrderDispatched.
37     }
38
39     function delayInManufacturing(uint orderId, uint _delayTime) onlyOwner{
40
41         uint delayTime = _delayTime;
42         statsMap[orderId].checkPoint="OrderDelayed";
43         statsMap[orderId].timeTheEventCalled=now; // Updates currentStatusOfOrder.
44

```


TRACKING APPLICATION USING BLOCKCHAIN

```
Secure | https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.24+commit.e67f0147.js

browser/Manufacturer.sol browser/Customs.sol browser/LandTransport.sol browser/Escrow.sol x
ContractDefinition Escrow 0 reference(s) v

browser
config

1 pragma solidity ^0.4.11;
2
3 contract Escrow {
4     uint balance;
5     address public buyer;
6     address public seller;
7     address private escrow;
8     uint private start;
9     bool buyerOk;
10    bool sellerOk;
11
12    function Escrow(address buyer_address, address seller_address) public {
13        // this is the constructor function that runs ONCE upon initialization
14        buyer = buyer_address;
15        seller = seller_address;
16        escrow = msg.sender;
17        start = now; //now is an alias for block.timestamp, not really "now"
18    }
19
20    function accept() public {
21        if (msg.sender == buyer){
22            buyerOk = true;
23        } else if (msg.sender == seller){
24            sellerOk = true;
25        }
26        if (buyerOk && sellerOk){
27            payBalance();
28        } else if (buyerOk && !sellerOk && now > start + 30 days) {
29            // Freeze 30 days before release to buyer. The customer has to remember to call this method after freeze period.
30            selfdestruct(buyer);
31        }
32    }
33
34    function payBalance() private {
35        // we are sending ourselves (contract creator) a fee
36        escrow.transfer(this.balance / 100);
37        // send seller the balance
38        if (seller.send(this.balance)) {
39            balance = 0;
40        } else {
41            throw;
42        }
43    }
44 }
```

```
Secure | https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.24+commit.e67f0147.js

browser/Manufacturer.sol browser/Customs.sol browser/LandTransport.sol x

browser
config

1 pragma solidity 0.4.24;
2 import "./manager.sol";
3 contract LandTransport is main {
4
5     event OrderLoaded(uint _orderId); // Defined Events.
6     event OrderInTransit(uint _orderId);
7     event OrderUnloaded(uint _orderId);
8
9     function getDetails(uint _orderId, uint _timeToNextEntity){ // Taken details;
10        uint orderId;
11        orderId = _orderId;
12        statsMap[orderId].timeToNextEntity = now + _timeToNextEntity; // Took time required to transfer to next entity.
13    }
14
15    function orderLoaded(uint _orderId){
16        OrderLoaded(_orderId); // Event OrderLoaded.
17    }
18
19    function stateRequiredTimeToNextEntity(uint _orderId, uint _requiredTime){ // Give Estimate;
20        statsMap[_orderId].timeToNextEntity = _requiredTime;
21    }
22
23    function orderInTransit(uint _orderId){
24        orderInTransit(_orderId);
25    }
26
27    function orderUnloaded(uint _orderId){
28        OrderUnloaded(_orderId);
29    }
30 }
31 }
```

CONCLUSION AND FUTURE ENHANCEMENT

- ▶ We have designed the model of tracking application using blockchain.
- ▶ We have implemented a full working backend of our tracking system for a product where a customer can order the product and there will be link between all the nodes of contracts and manager will initiate the process and at last it will be delivered to the customer respectively.

-<https://www.ibm.com/blockchain/what-is-blockchain.html>

<https://www.youtube.com/watch?v=r0LsnzAe1Yg>

-<https://www.youtube.com/watch?v=vWnFei-ugT8>

-<https://cryptozombies.io/>

-<https://blockgeeks.com/guides/solidity/>

-

<https://medium.com/@robertvermeulen/learn-solidity-the-ethereum---smart-contract-programming-language-7f106fc26d6>

<http://solidity.readthedocs.io/en/v0.4.24/units-and-global-variables.html>

